PENGGUNA DAN HAK ISTIMEWA, TIPE PENGGUNA , MELIHAT AKUN PENGGUNA

5

OBJEKTIF :

1. Mahasiswa Mampu Berganti Akun dan Mampu Menjalankan Perintah Dengan Hak Istimewa.
2. Mahasiswa Mampu Mengidentifikasi Informasi File yang Berisi Data User Account.
3. Mahasiswa Mampu Mengidentifikasi Informasi User Account.

PENDAHULUAN

User account dirancang untuk memberikan keamanan pada sistem operasi Linux. Setiap orang di sistem harus masuk menggunakan *user account* yang memungkinkan orang tersebut mengakses file dan direktori tertentu atau melarang akses tersebut, dicapai dengan menggunakan izin file, yaitu izin file dan direktori yang diberikan oleh sistem kepada *user*, grup, dan semua orang yang masuk. Izin ini dapat diedit oleh *root user*.

User account juga termasuk dalam grup, yang juga dapat digunakan untuk menyediakan akses ke file / direktori. Setiap *user* termasuk dalam setidaknya satu grup (seringkali banyak) untuk memungkinkan user lebih mudah berbagi data yang disimpan dalam file dengan *user* lain.

Data user account dan grup disimpan dalam file *database*. Mengetahui ini memungkinkan Anda untuk lebih memahami user mana yang memiliki akses ke file dan direktori di sistem. File *database* ini juga berisi informasi keamanan

penting yang dapat mempengaruhi kemampuan *user* untuk masuk dan mengakses sistem.

Beberapa perintah memberikan kemampuan untuk melihat informasi user account dan grup, serta untuk beralih dari satu *user account* ke akun lainnya (asalkan Anda memiliki kewenangan yang sesuai untuk melakukannya). Perintah ini berguna untuk menyelidiki useran sistem, memecahkan masalah sistem dan untuk memantau akses tidak sah ke sistem.

AKUN ADMIN

Ada banyak cara berbeda untuk menjalankan perintah yang membutuhkan hak akses administratif atau root. Masuk ke sistem sebagai *user root* memungkinkan Anda menjalankan perintah sebagai administrator. Akses ini berpotensi berbahaya karena Anda mungkin lupa bahwa Anda masuk sebagai *root* dan mungkin menjalankan perintah yang dapat menyebabkan masalah pada sistem. Akibatnya, tidak disarankan untuk masuk sebagai *user root* secara langsung.

Karena menggunakan akun *root* berpotensi berbahaya, Anda sebaiknya hanya menjalankan perintah sebagai *root* jika hak administratif diperlukan. Jika akun *root* dinonaktifkan, seperti pada distribusi Ubuntu, maka perintah administratif dapat dijalankan menggunakan perintah `sudo`. Jika akun *root* diaktifkan, maka user biasa dapat menjalankan perintah `su` untuk mengalihkan akun ke akun *root*.

Saat Anda masuk ke sistem secara langsung sebagai *root* untuk menjalankan perintah, maka segala sesuatu tentang sesi Anda akan dijalankan sebagai *user root*. Jika menggunakan lingkungan grafis, ini sangat berbahaya karena proses *login* grafis terdiri dari banyak file yang dapat dieksekusi (program yang dijalankan selama *login*). Setiap program yang berjalan sebagai *user root* mewakili ancaman yang lebih besar daripada proses yang dijalankan sebagai user standar, karena program tersebut akan diizinkan untuk melakukan hampir semua

hal, sedangkan program *user* standar sangat dibatasi dalam apa yang dapat mereka lakukan.

Bahaya potensial lainnya dengan masuk ke sistem sebagai *root* adalah bahwa orang yang melakukan ini mungkin lupa keluar untuk melakukan pekerjaan non-administratif mereka, memungkinkan program seperti browser dan klien email untuk dijalankan sebagai user root tanpa batasan pada apa. bisa mereka lakukan. Fakta bahwa beberapa distribusi Linux, terutama Ubuntu, tidak mengizinkan user untuk masuk karena *user root* seharusnya menjadi indikasi yang cukup bahwa ini bukan cara yang disukai untuk melakukan tugas administratif.

BERALIH PENGGUNA

Perintah `su` memungkinkan Anda menjalankan *shell* sebagai *user* yang berbeda. Saat beralih ke *user* root adalah perintah `su` adalah yang paling sering digunakan, perintah `su` juga dapat beralih ke user lain.

```
su [options] [username]
```

Saat berpindah *user* menggunakan opsi *shell login* direkomendasikan, karena *shell login* sepenuhnya mengkonfigurasi *shell* baru dengan pengaturan *user* baru, memastikan semua perintah yang dijalankan berjalan dengan benar. Jika opsi ini dihilangkan, *shell* baru akan mengubah UID tetapi tidak sepenuhnya memasukkan *user*. Opsi *shell login* dapat ditentukan dengan salah satu dari tiga cara:

```
su -

su -l

su --login
```

Secara default, jika nama user tidak ditentukan, perintah su membuka *shell* baru sebagai *user root*. Dua perintah berikut adalah cara yang setara untuk memulai *shell* sebagai *user root*:

```
su - root

su -
```

Setelah menekan **Enter** untuk menjalankan salah satu dari perintah ini, *user* harus memberikan *password user root* untuk memulai *shell* baru. Jika Anda tidak mengetahui *password* akun yang Anda tuju, maka perintah `su` akan gagal.

Perhatikan pada contoh di bawah ini, dan di mesin virtual kami, prompt perintah berubah untuk menampilkan user saat ini.

```
sysadmin@localhost:~$ su -

Password: netlab123

root@localhost:~# id

uid=0(root) gid=0(root) groups=0(root)
```

Setelah menggunakan *shell* yang dimulai dengan perintah `su` untuk melakukan tugas administratif yang diperlukan, kembali ke shell asli Anda (dan user account asli) dengan menggunakan perintah `exit`.

```
root@localhost:~# exit

logout

sysadmin@localhost:~$ id

uid=1001(sysadmin) gid=1001(sysadmin) groups=1001(sysadmin),4(adm)
,27(sudo)
```

MELAKUKAN PERINTAH HAK ISTIMEWA

Perintah `sudo` memungkinkan *user* untuk menjalankan perintah sebagai user lain. Mirip dengan perintah `su`, *user root* diasumsikan secara default.

```
sudo [options] command
```

Dalam distribusi yang tidak mengizinkan *user root* untuk masuk secara langsung atau melalui perintah su, proses instalasi secara otomatis mengkonfigurasi satu *user account* agar dapat menggunakan perintah sudo untuk menjalankan perintah seolah-olah user root menjalankannya. Misalnya, hak akses administratif diperlukan untuk melihat file /etc/shadow:

```
sysadmin@localhost:~$ head /etc/shadow
head: cannot open '/etc/shadow' for reading: Permission denied
```

Saat menggunakan perintah sudo untuk menjalankan perintah sebagai *user root*, perintah tersebut meminta password user itu sendiri, bukan password *user root*. Fitur keamanan ini dapat mencegah akses administratif yang tidak sah jika *user* meninggalkan komputer mereka tanpa pengawasan. Perintah untuk password tidak akan muncul lagi selama user terus menjalankan perintah sudo kurang dari lima menit.

Perintah sudo berikut akan menjalankan perintah head dari contoh sebelumnya sebagai *user root*. Ini meminta password user sysadmin:

```
sysadmin@loc.0alhost:~$ sudo head /etc/shadow
[sudo] password for sysadmin: netlab123
root:$6$4Yga95H9$8HbxqsMEIBTZ0YomlMffYCV9VE1SQ4T2H3SHXw41M02SQtfAd
DVE9mqGp2hr20q.ZuncJpLyWkYwQdKlSJyS8.:16464:0:99999:7:::
daemon:*:16463:0:99999:7:::
bin:*:16463:0:99999:7:::
sys:*:16463:0:99999:7:::
sync:*:16463:0:99999:7:::
games:*:16463:0:99999:7:::
man:*:16463:0:99999:7:::
```

```
lp:*:16463:0:99999:7:::

mail:*:16463:0:99999:7:::

news:*:16463:0:99999:7:::
```

Menggunakan perintah sudo untuk menjalankan perintah administratif akan menghasilkan entri yang ditempatkan di file *log*. Setiap entri menyertakan nama user yang mengeksekusi perintah, perintah yang dijalankan serta tanggal dan waktu eksekusi. Hal ini memungkinkan peningkatan akuntabilitas, dibandingkan dengan sistem di mana banyak user mungkin mengetahui password root dan dapat masuk secara langsung sebagai root atau menggunakan perintah su untuk menjalankan perintah sebagai user root.

Satu keuntungan besar menggunakan sudo untuk menjalankan perintah administratif adalah mengurangi risiko user secara tidak sengaja menjalankan perintah sebagai root. Tujuan untuk menjalankan perintah sudah jelas; perintah tersebut dijalankan sebagai root jika diawali dengan perintah sudo. Jika tidak, perintah akan dijalankan sebagai user biasa.

AKUN PENGGUNA

Ada beberapa file teks di direktori /etc yang berisi data *user account* dan grup yang ditentukan di sistem. Misalnya, untuk melihat apakah user account tertentu telah ditentukan di sistem, maka tempat untuk memeriksanya adalah file /etc/passwd.

File /etc/passwd mendefinisikan beberapa informasi akun untuk user account. Contoh berikut menunjukkan lima baris terakhir dari file /etc/passwd:

```
sysadmin@localhost:~$ tail -5 /etc/passwd

syslog:x:101:103::/home/syslog:/bin/false

bind:x:102:105::/var/cache/bind:/bin/false

sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
```

```
operator:x:1000:37::/root:/bin/sh

sysadmin:x:1001:1001:System Administrator,,,,:/home/sysadmin:/bin/bash
```

Setiap baris berisi informasi yang berkaitan dengan satu user. Data dipisahkan menjadi beberapa field dengan karakter titik dua (:). Berikut ini menjelaskan setiap field secara rinci, dari kiri ke kanan, menggunakan baris terakhir dari keluaran grafik sebelumnya:

Nama

```
sysadmin:x:1001:1001:System Administrator,,,,:/home/sysadmin:/bin/bash
```

Kolom pertama berisi nama user atau nama user. Nama ini digunakan saat masuk ke sistem dan saat kepemilikan file dilihat dengan perintah `ls -l`. Ini disediakan untuk memudahkan user biasa merujuk ke akun tersebut, sementara sistem biasanya menggunakan ID user secara internal.

Penampung Kata Sandi

```
sysadmin:x:1001:1001:System Administrator,,,,:/home/sysadmin:/bin/bash
```

Pada suatu waktu, password untuk user disimpan di lokasi ini, namun, sekarang x di Field ini menunjukkan kepada sistem bahwa password ada di file /etc/shadow.

ID Pengguna

```
sysadmin:x:1001:1001:System Administrator,,,,:/home/sysadmin:/bin/bash
```

Setiap akun diberi ID user (UID). Nama user tidak digunakan secara langsung oleh sistem, yang biasanya mendefinisikan akun oleh UID. Misalnya, file dimiliki oleh UID, bukan oleh nama user.

ID Grup Utama

```
sysadmin:x:1001:1001:System Administrator,,,,:/home/sysadmin:/bin
/bash
```

Field ini menunjukkan bahwa user adalah anggota grup itu, yang berarti user memiliki izin khusus pada file apa pun yang dimiliki oleh grup ini.

Komentar

```
sysadmin:x:1001:1001:System Administrator,,,,:/home/sysadmin:/bin
/bash
```

Field ini dapat berisi informasi apa pun tentang user, termasuk nama asli mereka atau informasi berguna lainnya.

Direktori Home

```
sysadmin:x:1001:1001:System Administrator,,,,:/home/sysadmin:/bin
/bash
```

Field ini menentukan lokasi direktori home user. Untuk user biasa, biasanya ini adalah /home/*username*. Misalnya, nama user bob akan memiliki direktori home /home/bob.

User root biasanya memiliki tempat berbeda untuk direktori home, direktori /root.

Shell

```
sysadmin:x:1001:1001:System Administrator,,,,:/home/sysadmin:/bin
/bash
```

Field ini menunjukkan lokasi shell login user. Secara default, user ditempatkan di shell ini setiap kali mereka masuk ke lingkungan baris perintah atau membuka jendela terminal. Bash shell /bin/bash adalah shell paling umum untuk user Linux.

Cara yang efisien untuk memeriksa apakah user tertentu telah didefinisikan pada sistem adalah dengan mencari file /etc/passwd menggunakan perintah grep. Misalnya, untuk melihat informasi user account bernama sysadmin, gunakan perintah berikut:

```
sysadmin@localhost:~$ grep sysadmin /etc/passwd
sysadmin:x:1001:1001:System Administrator,,,,:/home/sysadmin:/bin/
bash
```

KATA SANDI

Seperti yang disebutkan sebelumnya, file /etc/shadow berisi informasi akun yang terkait dengan *password user*. Namun, *user* biasa tidak dapat melihat konten file /etc/shadow untuk alasan keamanan. Untuk melihat konten file ini, *login* sebagai administrator (*root*):

```
sysadmin@localhost:~$ su -
Password: netlab123
root@localhost:~#
```

File /etc/shadow biasa akan terlihat seperti berikut:

```
root@localhost:~# tail -5 /etc/shadow
```

```
syslog:*:16874:0:99999:7:::

bind:*:16874:0:99999:7:::

sshd:*:16874:0:99999:7:::

operator:!:16874:0:99999:7:::

sysadmin:$6$c75ekQWF$.GpiZpFnIXLzkALjDpZXmjxZcIll14OvL2mFSIfnc1aU2
cQ/221QL5AX5RjKXpXPJRQ0uVN35TY3/..c7v0.n0:16874:5:30:7:60:15050:
```

Sekali lagi, setiap baris dipisahkan menjadi beberapa *field* dengan karakter titik dua. Berikut ini menjelaskan setiap *field* secara rinci, dari kiri ke kanan, menggunakan baris terakhir dari keluaran grafik sebelumnya:

Nama Pengguna

```
sysadmin:$6$c75ekQWF$.GpiZpFnIXLzkALjDpZXmjxZcIll14OvL2mFSIfnc1a
U2cQ/221QL5AX5RjKXpXPJRQ0uVN35TY3/..c7v0.n0:16874:5:30:7:60:1505
0::
```

Field password berisi password terenkripsi untuk akun tersebut. String yang sangat panjang ini adalah enkripsi satu arah, artinya tidak dapat "dibalik" untuk menentukan sandi asli.

Meskipun user biasa memiliki sandi terenkripsi di field ini, akun sistem memiliki karakter * (*asterisk*) di field ini.

Terakhir Diubah

```
sysadmin:$6$c75ekQWF$.GpiZpFnIXLzkALjDpZXmjxZcIll14OvL2mFSIfnc1aU2
cQ/221QL5AX5RjKXpXPJRQ0uVN35TY3/..c7v0.n0:16874:5:30:7:60:15050:
:
```

Field ini berisi angka yang mewakili terakhir kali password diubah. Angka 16874 adalah jumlah hari sejak 1 Januari 1970 (disebut *Epoch*). Nilai ini dihasilkan

secara otomatis ketika password user diubah. Ini digunakan oleh fitur penuaan password yang disediakan oleh sisa field file ini.

Minimum

```
sysadmin:$6$c75ekQWF$.GpiZpFnIXLzkALjDpZXmjxZcIll14OvL2mFSIfnc1aU2
cQ/221QL5AX5RjKXpXPJRQ0uVN35TY3/..c7v0.n0:16874:5:30:7:60:15050:
:
```

Field ini menunjukkan jumlah hari ***minimum*** antara perubahan password. Ini adalah salah satu field penuaan password; nilai bukan nol di field ini menunjukkan bahwa setelah user mengubah sandi mereka, sandi tidak dapat diubah lagi selama jumlah hari yang ditentukan, 5 hari dalam Field ini. Field ini penting ketika field ***maksimum*** digunakan.

Nilai nol di field ini berarti user selalu dapat mengubah sandi mereka.

Maximum

```
sysadmin:$6$c75ekQWF$.GpiZpFnIXLzkALjDpZXmjxZcIll14OvL2mFSIfnc1aU2
cQ/221QL5AX5RjKXpXPJRQ0uVN35TY3/..c7v0.n0:16874:5:30:7:60:15050:
```

Field ini menunjukkan jumlah hari **maksimum** password valid. Ini digunakan untuk memaksa user mengubah password mereka secara teratur. Nilai 30 di field ini berarti user harus mengubah sandi mereka setidaknya setiap 30 hari untuk menghindari akun mereka terkunci.

Perhatikan bahwa jika kolom ***minimum*** disetel ke 0, user mungkin dapat segera menyetel sandi mereka kembali ke nilai aslinya, mengalahkan tujuan memaksa user untuk mengubah sandi mereka setiap 30 hari. Jadi, jika field ***maksimum*** disetel, field minimum biasanya juga disetel.

Misalnya, *minimum: maksimum* 5:30berarti user harus mengubah sandi mereka setiap 30 hari dan, setelah mengubah, user harus menunggu 5 hari sebelum mereka dapat mengubah sandi mereka lagi.

Jika field max disetel ke 99999, nilai maksimum yang mungkin, maka pada dasarnya user tidak perlu mengubah sandi mereka (karena 99999 hari kira-kira 274 tahun).

Peringatan

```
sysadmin:$6$c75ekQWF$.GpiZpFnIXLzkALjDpZXmjxZcIll14OvL2mFSIfnc1aU2
cQ/221QL5AX5RjKXpXPJRQ0uVN35TY3/..c7v0.n0:16874:5:30:7:60:15050:
:
```

Jika field maksimum disetel, field peringatan menunjukkan jumlah hari sebelum password kedaluwarsa yang diperingatkan oleh sistem kepada user. Misalnya, jika field peringatan diatur ke 7, maka kapan saja selama 7 hari sebelum jangka waktu maksimum tercapai, user akan diperingatkan untuk mengubah password mereka selama proses masuk.

User hanya diperingatkan saat masuk, jadi beberapa administrator telah mengambil pendekatan untuk menyetel field peringatan ke nilai yang lebih tinggi untuk memberikan kesempatan yang lebih besar untuk mengeluarkan peringatan.

Jika kerangka waktu maksimum diatur ke 99999, maka field peringatan pada dasarnya tidak berguna.

Tidak Aktif

```
sysadmin:$6$c75ekQWF$.GpiZpFnIXLzkALjDpZXmjxZcIll14OvL2mFSIfnc1aU2
cQ/221QL5AX5RjKXpXPJRQ0uVN35TY3/..c7v0.n0:16874:5:30:7:60:15050:
:
```

Jika user mengabaikan peringatan dan melebihi jangka waktu sandi, akun mereka akan terkunci. Dalam hal ini, field tidak aktif memberi user masa "tenggang" di mana sandi mereka dapat diubah, tetapi hanya selama proses masuk.

Jika field tidak aktif disetel ke 60, user memiliki waktu 60 hari untuk mengubah ke password baru. Jika mereka gagal melakukannya, maka administrator akan diperlukan untuk mengatur ulang password untuk user tersebut.

Berakhir

```
sysadmin:$6$c75ekQWF$.GpiZpFnIXLzkALjDpZXmjxZcIll14OvL2mFSIfnc1aU2
cQ/221QL5AX5RjKXpXPJRQ0uVN35TY3/..c7v0.n0:16874:5:30:7:60:15050:
:
```

Field ini menunjukkan hari akun akan kedaluwarsa, yang diwakili oleh jumlah hari sejak 1 Januari 1970. Akun yang kedaluwarsa dikunci, tidak dihapus, artinya administrator dapat mengatur ulang password untuk membuka kunci akun.

Akun dengan tanggal kedaluwarsa biasanya diberikan kepada karyawan atau kontraktor sementara. Akun secara otomatis kedaluwarsa setelah hari terakhir user bekerja.

Ketika administrator menyetel field ini, alat digunakan untuk mengubah dari tanggal sebenarnya ke tanggal Epoch. Ada juga beberapa konverter gratis yang tersedia di Internet.

Disimpan

```
sysadmin:$6$c75ekQWF$.GpiZpFnIXLzkALjDpZXmjxZcIll14OvL2mFSIfnc1aU2
cQ/221QL5AX5RjKXpXPJRQ0uVN35TY3/..c7v0.n0:16874:5:30:7:60:15050:
:
```

Saat ini tidak digunakan, field ini dicadangkan untuk useran di masa mendatang.

Perlu diingat!

Selain perintah grep, teknik lain untuk mengambil informasi user yang terdapat dalam file /etc/passwd dan /etc/shadow adalah dengan menggunakan perintah getent. Satu keuntungan dari perintah ini adalah dapat mengambil informasi akun yang ditentukan secara lokal, dalam file seperti /etc/passwd dan /etc/shadow, atau pada server direktori jaringan.

Sintaks umum dari perintah getent adalah:

```
getent database record
```

Misalnya, perintah berikut akan mengambil informasi akun untuk user sysadmin dari file /etc/passwd:

```
sysadmin@localhost:~$ getent passwd sysadmin
sysadmin:x:1001:1001:System Administrator,,,,:/home/sysadmin:/bin/bash
```

AKUN SISTEM

User masuk ke sistem menggunakan user account biasa. Biasanya, akun ini memiliki nilai UID lebih dari 500 (pada beberapa sistem 1.000). User root memiliki akses khusus ke sistem. Akses ini diberikan ke akun dengan UID 0.

Ada akun tambahan yang tidak dirancang bagi user untuk masuk. Akun ini, biasanya dari UID 1 hingga UID 499, disebut *system account,* dan mereka dirancang untuk menyediakan akun untuk layanan yang berjalan pada sistem.

System account memiliki beberapa field di file /etc/passwd dan /etc/shadow yang berbeda dari akun lain. Misalnya, system account jarang memiliki direktori home karena biasanya tidak digunakan untuk membuat atau menyimpan file. Di file /etc/passwd, system account memiliki program non-login di field *shell*:

```
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
```

Di /etc/shadow, system account biasanya memiliki karakter asterisk * sebagai pengganti field password:

```
sshd:*:16874:0:99999:7:::
```

Sebagian besar system account diperlukan agar sistem berfungsi dengan benar. Anda tidak boleh menghapus system account kecuali Anda yakin bahwa menghapus akun tidak akan menimbulkan masalah. Luangkan waktu untuk mempelajari apa yang dilakukan setiap system account; administrator sistem ditugaskan untuk memastikan keamanan sistem, dan itu termasuk mengamankan *system account* dengan benar.

Akun Grup

Tingkat akses Anda ke sistem tidak hanya ditentukan oleh *user account* Anda. Setiap *user* dapat menjadi anggota dari satu atau beberapa grup, yang juga dapat memengaruhi tingkat akses ke sistem.

Biasanya, sistem UNIX membatasi user untuk menjadi anggota tidak lebih dari total enam belas grup, tetapi kernel Linux terkini mendukung user dengan lebih dari enam puluh lima ribu keanggotaan grup.

File /etc/passwd mendefinisikan keanggotaan grup utama untuk user. Keanggotaan grup tambahan (atau keanggotaan grup sekunder) dan grup itu sendiri ditentukan dalam file /etc/group.

File /etc/group adalah file lain yang dipisahkan oleh titik dua. Berikut ini menjelaskan field secara lebih detail, menggunakan baris yang menjelaskan akun grup pada umumnya.

Nama Grup

```
mail:x:12:mail,postfix
```

Field ini berisi nama grup. Seperti nama *user*, nama lebih mudah diingat orang daripada angka. Sistem biasanya menggunakan ID grup daripada nama grup.

Penampung Kata Sandi

```
mail:x:12:mail,postfix
```

Meskipun ada password untuk grup, password jarang digunakan di Linux. Jika administrator membuat password grup, itu akan disimpan di file /etc/shadow. Tanda x di field ini digunakan untuk menunjukkan bahwa password tidak disimpan dalam file ini.

GID

```
mail:x:12:mail,postfix
```

Setiap grup dikaitkan dengan unique group ID (GID) yang ditempatkan di field ini.

Daftar Pengguna

```
mail:x:12:mail,postfix
```

Field terakhir ini digunakan untuk menunjukkan siapa yang menjadi anggota grup. Sementara keanggotaan grup utama ditentukan di file /etc/passwd, user yang ditugaskan ke grup tambahan akan memiliki nama *user* mereka ditempatkan di field ini dari file /etc/group. Dalam kasus ini, user email dan postfix adalah anggota sekunder dari grup email.

Nama *user* juga sering muncul sebagai nama grup. Juga biasa bagi user untuk menjadi bagian dari grup dengan nama yang sama.

Untuk melihat informasi tentang grup tertentu, perintah `grep` atau `getent` dapat digunakan. Misalnya, perintah berikut menampilkan informasi akun grup email:

```
sysadmin@localhost:~$ grep mail /etc/group

mail:x:12:mail,postfix

sysadmin@localhost:~$ getent group mail

mail:x:12:mail,postfix
```

MELIHAT INFORMASI PENGGUNA

Perintah `id` digunakan untuk mencetak informasi pengguna dan grup untuk pengguna tertentu.

```
id [options] username
```

Saat beralih di antara akun pengguna yang berbeda, dapat membingungkan akun mana yang saat ini masuk. Ketika dijalankan tanpa argumen, perintah `id` mengeluarkan informasi tentang pengguna saat ini, memungkinkan Anda untuk mengkonfirmasi identitas Anda pada sistem.

```
sysadmin@localhost:~$ id

uid=1001(sysadmin) gid=1001(sysadmin) groups=1001(sysadmin),4(adm)
,27(sudo)
```

Output dari perintah `id` selalu mencantumkan informasi akun pengguna terlebih dahulu, menggunakan ID pengguna dan nama pengguna terlebih dahulu:

```
uid=1001(sysadmin) gid=1001(sysadmin) groups=1001(sysadmin),4(adm),27(sudo)
```

Setelah nama pengguna, grup utama terdaftar, dilambangkan dengan ID grup dan nama grup:

```
uid=1001(sysadmin) gid=1001(sysadmin) groups=1001(sysadmin),4(adm),27(sudo)
```

Informasi lain yang terdaftar termasuk grup milik pengguna, lagi-lagi dilambangkan dengan ID grup diikuti dengan nama grup. Pengguna yang ditampilkan termasuk dalam tiga grup:

```
uid=1001(sysadmin) gid=1001(sysadmin) groups=1001(sysadmin),4(adm),27(sudo)
```

Jika perintah diberi nama pengguna sebagai argumen, seperti root, ini akan menampilkan informasi tentang akun yang ditentukan:

```
sysadmin@localhost:~$ id root
uid=0(root) gid=0(root) groups=0(root)
```

Untuk mencetak hanya grup utama pengguna, gunakan opsi -g:

```
sysadmin@localhost:~$ id -g
1001
```

Perintah id juga dapat digunakan untuk memverifikasi keanggotaan grup sekunder pengguna, untuk mencetak informasi ini, gunakan opsi -G:

```
sysadmin@localhost:~$ id -G

1001 4 27
```

Output dari contoh sebelumnya sejajar dengan konten file /etc/group, seperti yang ditunjukkan oleh pencarian sysadmin:

```
sysadmin@localhost:~$ cat /etc/group | grep sysadmin

adm:x:4:syslog,sysadmin

sudo:x:27:sysadmin

sysadmin:x:1001:
```

MELIHAT PENGGUNA SAAT INI

Perintah who menampilkan daftar pengguna yang saat ini masuk ke sistem, dari mana mereka masuk, dan kapan mereka masuk. Melalui penggunaan option, perintah ini juga dapat menampilkan informasi seperti runlevel saat ini (sebuah status fungsional komputer) dan waktu sistem di-boot.

Sebagai contoh:

```
sysadmin@localhost:~$ who

root             tty2         2013-10-11 10:00

sysadmin         tty1         2013-10-11 09:58 (:0)

sysadmin         pts/0        2013-10-11 09:59 (:0.0)

sysadmin         pts/1        2013-10-11 10:00 (example.com)
```

Berikut ini menjelaskan keluaran dari perintah who:

Username

```
root             tty2         2013-10-11 10:00
```

Kolom ini menunjukkan nama pengguna yang login. Perhatikan bahwa "login" yang kami maksud adalah "proses login dan open terminal window ".

Terminal

```
root             tty2         2013-10-11 10:00
sysadmin         pts/0        2013-10-11 09:59 (:0.0)
```

Kolom ini menunjukkan jendela terminal mana yang sedang digunakan pengguna.

Jika nama terminal dimulai dengan tty, maka ini adalah terminal dimulai dengan tty, maka ini adalah indikasi login lokal, karena ini adalah terminal baris perintah biasa. Jika nama terminal dimulai dengan pts, maka ini menunjukkan bahwa pengguna menggunakan terminal semu atau menjalankan proses yang bertindak sebagai terminal.

Date

```
root             tty2         2013-10-11 10:00
```

Kolom ini menunjukkan kapan pengguna masuk.

Host

Setelah tanggal dan waktu, beberapa informasi lokasi mungkin muncul. Jika informasi lokasi berisi nama host, nama domain, atau alamat IP, maka pengguna telah masuk dari jarak jauh:

```
sysadmin        pts/1        2013-10-11 10:00 (example.com)
```

Jika ada titik dua dan angka, maka ini menunjukkan bahwa mereka telah melakukan login grafis lokal:

```
sysadmin      tty1         2013-10-11 09:59 (:0)
```

Jika tidak ada informasi lokasi yang ditampilkan di kolom terakhir, ini berarti pengguna masuk melalui proses baris perintah lokal:

```
root             tty2         2013-10-11 10:00
```

Perlu diingat!

Perintah who memiliki beberapa opsi untuk menampilkan informasi status sistem. Misalnya, opsi -b menunjukkan terakhir kali sistem dimulai (boot), dan opsi -r menunjukkan waktu sistem mencapai run level saat ini:

```
sysadmin@localhost:~$ who -b -r
         system boot   2013-10-11 09:54
         run-level 5    2013-10-11 09:54
```

Mungkin ada contoh di mana informasi lebih lanjut tentang pengguna, dan apa yang mereka lakukan di sistem, diperlukan. Perintah w memberikan daftar yang lebih rinci tentang pengguna saat ini di sistem daripada perintah who. Ini juga memberikan ringkasan status sistem. Sebagai contoh:

```
sysadmin@localhost:~$ w
 10:44:03 up 50 min,  4 users,  load average: 0.78, 0.44, 0.19
USER            TTY      FROM        LOGIN@   IDLE      JCPU    PCPU
WHAT
root            tty2     -            10:00    43:44     0.01s   0.01s
-bash
sysadmin        tty1     :0           09:58    50:02     5.68s   0.16s
pam: gdm-password
sysadmin        pts/0    :0.0         09:59    0.00s       0.14s
        0.13s    ssh 192.168.1.2
sysadmin        pts/1    example.com 10:00    0.00s     0.03s   0.01s
w
```

Baris pertama keluaran dari perintah w identik dengan perintah uptime. Ini menunjukkan waktu saat ini, berapa lama sistem telah berjalan, jumlah total pengguna yang saat ini masuk dan beban pada sistem rata-rata selama periode waktu 1, 5 dan 15 menit terakhir. Rata-rata beban adalah penggunaan CPU di mana nilai 100 berarti penggunaan CPU penuh selama periode waktu tersebut.

Berikut ini menjelaskan sisa output dari perintah w:

Kolom	Contoh	Deskripsi
USER	root	Nama pengguna yang masuk
TTY	tty2	Jendela terminal mana tempat pengguna bekerja.

Kolom	Contoh	Deskripsi
FROM	example.com	Dari mana pengguna masuk.
LOGIN@	10:00	Saat pengguna masuk.
IDLE	43:44	Berapa lama pengguna tidak aktif sejak perintah terakhir dijalankan.
JCPU	0.01s	Total waktu cpu yang digunakan oleh semua proses berjalan sejak login.
PCPU	0.01s	Total waktu cpu untuk proses saat ini.
WHAT	-bash	Proses saat ini yang sedang dijalankan pengguna.

Catatan: Karakter s mengartikan detik.

MELIHAT RIWAYAT LOGIN

Perintah last membaca seluruh riwayat login dari file /var/log/wtmp dan menampilkan semua login dan catatan reboot secara default. Detail yang menarik dari catatan reboot adalah bahwa ia menampilkan versi kernel Linux yang di-boot,

bukan lokasi login. File /var/log/wtmp menyimpan log dari semua pengguna yang telah masuk dan keluar dari sistem.

```
sysadmin@localhost:~$ last

sysadmin console Tue Sep 18 02:31   still logged in

sysadmin console                    Tue Sep 18 02:31 - 02:31  (00:
00)

wtmp begins Tue Sep 18 02:31:57 2018
```

Perintah last sedikit berbeda dari perintah who dan w. Secara default, ini juga menunjukkan nama pengguna, terminal, dan lokasi login, tidak hanya sesi login saat ini, tetapi juga sesi sebelumnya. Berbeda dengan perintah who dan w, ini menampilkan tanggal dan waktu pengguna masuk ke sistem. Jika pengguna telah keluar dari sistem, maka itu akan menampilkan total waktu yang dihabiskan pengguna untuk masuk, jika tidak maka akan ditampilkan masih masuk.

Perlu diingat!

Perintah who membaca dari file /var/log/utmp yang mencatat pengguna saat ini, sedangkan perintah last membaca dari file /var/log/wtmp, yang menyimpan riwayat semua login pengguna.